

**Survey and Analysis of the National Security
High Performance Computing Architectural Requirements**

4 June 2001

Richard Games, Ph.D.
rg@mitre.org

The MITRE Corporation
Bedford, MA 01730

Approved for public release; distribution unlimited

Review of this Material does not imply Department of Defense
Endorsement of Factual Accuracy or Opinion.

EXECUTIVE SUMMARY

Are current high performance computers (HPCs) that use commodity microprocessors (“commodity HPCs”) adequate for national security applications? Or is there a critical need for traditional vector supercomputers? These were the overarching questions of a quick-reaction survey conducted by the author during the first two weeks of April 2001. The Deputy Under Secretary of Defense for Science and Technology commissioned the survey to help understand the national security issues associated with Cray Inc.’s request to the United States Department of Commerce to lift the import duty on Japanese vector supercomputers. This report contains the analysis of the survey results along with recommendations.

There are no simple answers to these two questions. At the minimum the answers depend on the specific application area. The author conducted face-to-face interviews with multiple users and software developers in 10 high performance computing application areas, including support to research and development, acquisition, and operations. Operational support ranged from “off-line” predictive analysis for planning purposes to “on-line” applications such as weather prediction, surveillance, and reconnaissance.

Based on the interviews conducted, the overall assessment was that commodity HPCs are providing useful capability in all areas surveyed except for the cryptanalysis area. Over the last five years there has been a major investment made by the national security community to retool legacy vector-supercomputer software to run on commodity HPCs. As a result, these systems are in high demand by the national security community.

Although commodity HPCs are producing useful results, in almost every case there were significant issues identified with their use. These issues included the negative impact that their difficult programming environments are having on researchers and system developers and the inefficiency in many cases of the actual processing due to a serious processor-memory communications bottleneck. The real value of this survey came from the insights gained from the discussions of the challenges that national security users and software developers face because of the current HPC technology base.

The bottom line for the national security community reduces to the interrelated issues of *productivity* and *affordability*. How productive are the researchers/developers that write the high performance software? How productive (efficient) are the HPCs that run the software? What does it cost? Included in the total cost are the facilities and the operational risks associated with the reliability of larger less-efficient installations. Perhaps the hardest cost to quantify is the “opportunity lost” when a domain researcher spends time on complicated computer programming rather than on creating new science.

A number of recommendations are made to increase the flexibility and performance of future national security HPC options. These include assessing the impact of current state-of-the-art Japanese vector supercomputers, promulgating the software best practices identified during the survey, and initiating a pragmatic R&D program to improve the productivity of HPCs for national security applications.

ACKNOWLEDGMENTS

John Grosh, Charles Holland, and Mark Norton from the Office of the Secretary of Defense provided valuable support during the course of the survey. Many members of the national security community who participated in the survey took time off from their busy schedules to discuss their current and past experiences with high performance computers.

1. INTRODUCTION

The national security community uses high performance computers (HPCs) as *tools* for performing research and development; for supporting the acquisition process, including test and evaluation; for training; and for operations. Operational support ranges from “off-line” predictive analysis for planning purposes to “on-line” applications such as weather prediction, signal and image processing for intelligence, surveillance, and reconnaissance (ISR), or as a component of a weapon system.

An HPC’s value depends on how well it performs the target applications. A number of performance factors are involved: delivering correct answers in the time desired/required (wall-clock time to solution), being reliable, being easy to program, being cost effective, and in some cases, meeting size, weight, and power constraints.

Computer architecture refers to how a system’s components such as processors, memory, internal networks, and input/output devices are combined into a single computing resource. The compatibility between the computer architecture and the target application is one factor that determines how well the HPC performs. This paper reports on the results of a “quick-reaction” survey to assess the compatibility of current commercial HPC architectures for performing national security applications.

The overarching question considered was whether current commercial HPCs that use commodity microprocessors are adequate for national security applications? Or whether there is a critical need for traditional vector supercomputers? The survey was motivated by the request by Cray Inc. to the Department of Commerce to lift the import duty on Japanese vector supercomputers¹. Cray plans to resell NEC vector supercomputers in the United States and elsewhere while it completes the development of its next-generation SV-2 vector supercomputer. The Department of Defense (DOD) is currently co-investing in the development of the SV-2.

The next section provides a brief overview of current HPC options and provides a brief historical context. Section 3 introduces performance metrics needed in the subsequent application assessments. Section 4 describes the survey methodology and lists the national-security applications covered. Section 5 summarizes the overall national-security application/HPC-architecture assessment. Section 6 discusses the general findings of the survey, and Section 7 concludes with recommendations. Appendix A contains the questions used to conduct the interviews. Appendix B contains more detailed information on the specific applications covered, including the organizations that participated in the survey. Appendix C contains a copy of the memorandum that initiated the study.

¹ The Commerce Department lifted the duty on 3 May 2001.

2. HPC BACKGROUND

For the purpose of this assessment, HPC architectures can be organized along the two dimensions shown in Figure 1: processor type and programming model supported. Systems based on custom vector processors will be referred to as *vector supercomputers*. These include systems from Cray Research (YMP, C90, T90, SV-1), Cray Inc. (SV-2), NEC (SX-5), and Fujitsu (VPP 5000).

Systems that incorporate commodity microprocessors will be referred to as *commodity HPCs*. These include systems from Compaq, Hewlett Packard, IBM, SGI, Sun, and a variety of Windows-Intel (WINTEL) vendors. In addition to servicing technical users, these companies compete in the desktop and server marketplace for database and internet applications where price/performance competition is especially keen. Clustering these desktop or small servers using commodity or high-performance networks and open source system software (e.g., Linux) is an important recent trend in this category intended to reduce the cost of high performance computing.

		Processor Type	
Programming Model		Custom Vector	Microprocessor
Shared-Memory Multi-Processing OpenMP		Parallel Vector Cray YMP, C90, T90, SV-1, SV-2, NEC SX-5	Symmetric Multiprocessors Distributed Shared Memory Compaq, HP, IBM, SGI, Sun, WINTEL
	Distributed-Memory Multi-Computing Message Passing Interface	Scalable Vector Cray SV-2, NEC SX-5, Fujitsu VPP 5000	Massively Parallel Processors Commodity Clusters Cray T3E, IBM SP3, SGI O3K
		Vector Supercomputer	Commodity HPC

Figure 1. High Performance Computer Architectures

The second dimension shown in Figure 1 concerns how parallel computers are programmed. In the *shared-memory* model, all processors share and have access to a common memory space, a single operating system schedules tasks on the multiple processors, and compiler technology exists to implement “loop-level” parallelism through programmer-supplied directives. This support reduces the complexity of parallel programming, but it is only effective on systems with a modest number of processors (generally 10s of processors).

The upper two quadrants of Figure 1 provide examples of both vector supercomputers and commodity HPCs that support the shared-memory model. The emergence of the OpenMP compiler directive standard makes it possible today to easily port shared-memory code amongst the various options (within each and horizontally between the upper two quadrants in Figure 1). However, architecture-specific software optimizations may still be required for either the vector processors or cache-based microprocessors.

In the *distributed-memory* model, each processor has its own memory space and operating system. Data in a remote processor's memory is accessed by explicitly sending messages between processors. Writing message-passing software can be complex, but can be effective on systems with a large number of processors (generally 100s to 1000s of processors).

The lower two quadrants in Figure 1 provide examples of both vector supercomputers and commodity HPCs that support the distributed-memory model. The emergence of the Message Passing Interface standard makes it easy to port distributed-memory code amongst the various options (within each and horizontally between the lower two quadrants in Figure 1). However, architecture-specific software optimizations may still be required for either the vector processors or cache-based microprocessors.

The current high-end computing situation can be summarized using Figure 1. Legacy HPC applications used compiler directives on shared-memory parallel vector supercomputers (upper left-hand quadrant). As commodity HPCs with distributed memory began to emerge in the early 1990s, applications began to be ported (or written from scratch) using message passing to run on these systems (lower right-hand quadrant).

These "diagonal" ports in Figure 1 were expensive multiyear software efforts and significantly increased the complexity of the HPC software base. By the mid-to-late 1990s the number of these "diagonal" porting activities increased as state-of-the-art vector supercomputers became unavailable in the US market. Today, the vast majority of high-end U.S. national-security software is based on the distributed-memory model, with a corresponding increase in software effort to maintain and evolve.

The capability of shared-memory commodity HPCs has improved considerably in the recent past. The appearance of distributed-memory hardware architectures that logically support the compiler-based shared-memory programming model (e.g., the SGI Origin and HP Exemplar) was an especially significant development. In many lower-end applications, this trend is obviating the need to switch to the distributed memory model.

3. PERFORMANCE METRICS

One way to assess how efficiently a computer is performing an application is to measure the rate that useful work is being performed. Often the application's workload can be measured in terms of the number of floating-point or fixed-point operations that need to be performed. Given the time to solution, it is then possible to calculate the average sustained processing rate. An efficiency metric is obtained by comparing this sustained processing rate to the theoretical peak-processing rate of the processor².

For simplicity, only floating-point operations (flop) will be considered in this report. If F equals the number of flop a computation requires, and if $t(1)$ is the time to solution (in seconds) on one processor, then the processor sustains on the average $S(1) = F/t(1)$ flop/s. If $P(1)$ is the theoretical peak flop/s rate of a single processor, then the *sustained processor utilization* for a single processor is defined as $u(1) = S(1)/P(1)$ (usually expressed as a percentage). A low sustained processor utilization can occur for a variety of reasons, including possibly a mismatch between the computer's architecture and the requirements of the application.

A second performance metric of interest is the *parallel efficiency*. Ideally, if the number of processors is doubled, then the time to solution should be halved. However, such perfect speedup is rarely achieved because of the lack of application parallelism or the overhead of communicating data between processors. If $t(n)$ is the time to solution when n processors are used, then the n processor speedup is defined as $speedup(n) = t(1)/t(n)$. Parallel efficiency for n processors is defined as $par_eff(n) = speedup(n)/n$ (usually expressed as a percentage). Parallel efficiency of less than 100 percent reduces the sustained processor utilization for the n -processor system: $u(n) = [F/t(n)]/[nP(1)] = u(1) \times par_eff(n)$.

Two kinds of parallel speedups are common:

1. *Fixed-problem* speedup: the problem size is fixed as the number of processors is increased. The decrease in the time to solution is measured. Fixed-problem speedup is the most common metric in operational uses where the time to solution must meet an external constraint.
2. *Scaled-problem* speedup: the problem size is increased as the number of processors is increased. The increase in the time to solution over the predicted time (usually constant) is measured. Scaled-problem speedup is often employed to assess R&D uses where "bigger (more resolution) is better," and machine size is only constrained by resource availability (in competition with other R&D users).

Both the sustained processor utilization and the parallel efficiency performance metrics are well known in the HPC community. However, obtaining data for sustained processor utilization during the course of the survey proved difficult as most users and some

² Users of HPCs are primarily interested in the time to solution and not this efficiency metric. However for the purpose of assessing architecture compatibility knowledge of this efficiency metric is essential.

developers interviewed do not keep track of this metric. On the other hand, parallel efficiencies (usually based on scaled-problem speedup) were easy to obtain and for the most part were quite high. But unfortunately a high “parallel efficiency” does not imply a parallel system is processing efficiently in terms of sustained processor utilization.

Many legacy HPC applications ran on shared-memory parallel vector supercomputers with sustained processor utilization in the range of 50 percent. On current commodity HPCs, the sustained processor utilization for these same applications often dropped to the range of 10 percent (or less) because of the current processor-memory communications bottleneck in these systems. The relative price per *sustained* flop/s (or Mflop/s or Gflop/s) for each of the processing options is crucial in determining which is ultimately more affordable.

4. SURVEY METHODOLOGY

The author conducted face-to-face interviews with HPC users and software developers from across the national-security community during the period 2-13 April 2001. DOD representatives scheduled most of the interviews. Some of the people interviewed were identified during the interview process.

Each interview covered the following topics:

1. Personal background
2. HPC applications and national security impact
3. Software description
4. Typical performance
5. Portability and porting
6. Current architectural limitations

Each topic was explored in depth using a series of questions that were developed in coordination with DOD representatives. Appendix A contains these questions.

The application areas covered were:

1. Operational weather and ocean forecasting
2. Planning activities for dispersion of airborne/waterborne contaminants
3. Engineering design of large aircraft, ship, and structures
4. Weapon (warheads and penetrators) effect studies
5. Improved armor design
6. Cryptanalysis
7. Survivability/stealthiness design
8. Intelligence, reconnaissance, and surveillance
9. National missile defense
10. Test and evaluation

Although a considerable number of application areas and research groups were involved in the two-week survey, the survey only represented a small (non-scientific) sampling by one person of the national security HPC community. Furthermore each application area could not be surveyed to the same depth. The number of people surveyed in each area is a rough indicator of the depth achieved. Finally, by design, the survey only involved groups from the DOD and Intelligence communities and did not involve the national security applications in the Department of Energy—although some of the codes considered are used by both communities.

5. SUMMARY ASSESSMENT

There are no simple answers to the two questions of whether current commodity HPCs are adequate for national security applications, or whether there is a critical need for traditional vector supercomputers. At the minimum the answer is application dependent. Figure 2 shows the overall assessments of the 10 application areas surveyed. A “red-yellow-green” indicator summarizes the assessment for each application area. Green indicates that commodity HPCs are providing needed capability. Yellow indicates that there are significant issues with the use of commodity HPCs, but generally they are providing the needed capability. Red indicates a serious problem with the use of commodity HPCs.

Assessment		<div> Ⓡ Serious problem with using commodity HPCs Ⓢ Some issues with using commodity HPCs Ⓤ Commodity HPCs providing capability </div>	
Overall	Applications	# People	Issues
Ⓤ	Operational weather and ocean forecasting	16	<div> Ⓡ Results not state of the art Ⓢ Dilution of research </div>
Ⓤ	Planning activities for dispersion of airborne/ waterborne contaminants (CBRN)	5	<div> Ⓡ Lack crises response Ⓢ Want access to vector SC </div>
Ⓤ	Engineering design of large aircraft, ship, and structures	6	<div> Ⓢ Actively seeking foreign access to vector SC </div>
Ⓤ	Weapon (warheads and penetrators) effect studies	3	<div> Ⓢ Grand challenges take months </div>
Ⓤ	Improved armor design	2	<div> Ⓢ Grand challenges take mths </div>
Ⓡ	Cryptanalysis	8	<div> Ⓢ Dilution of research </div>
Ⓤ	Survivability/stealthiness design	2	<div> Ⓢ New fast algorithms </div>
Ⓤ	Intelligence, surveillance, reconnaissance	5	<div> Ⓢ Accelerators needed </div>
Ⓤ	National missile defense	5	
Ⓤ	Test and Evaluation	2	<div> Ⓢ Accelerators needed </div>

Figure 2. Summary Assessment

Based on the interviews conducted, the overall assessment of each application area was judged to be “green” except for **cryptanalysis**, which was judged to be “red.” However, in every case but one there were significant issues identified with the use of commodity HPCs. Whether or not these individual issues “integrate” up to turn the overall area assessment from green to yellow to red is obviously a judgement that is subject to many factors. So it is important to look beyond the single color assigned to each application area. Rather the value of this survey comes from the insights gained from the one-on-one discussions of the challenges that national security users and software developers face because of the current HPC technology base. This section discusses these issues for each of the ten applications surveyed. Appendix B provides additional background information, including the organizations interviewed for each application area.

The **weather and ocean forecasting** application area is representative of what has happened in many computational fluid dynamics (CFD) applications over the last 10 years as researchers in the US have moved from the Cray vector supercomputers to commodity HPCs. The end result is that researchers, who were used to using Fortran codes that incorporated simple compiler directives for multitasking, have had to focus considerable time and attention on the programming complexities of distributed-memory MPI software development.

In the case of the weather and ocean forecasting groups interviewed, the transition from shared-memory to distributed-memory programming was very difficult and required significant resources over a three year period. These software porting projects were supported by the DOD HPC Modernization Program's Common High Performance Computing Software Support Initiative (CHSSI) and other Department of Energy and NASA sources. The efficiency of the converted codes has suffered because of the memory-processor bottleneck in commodity HPCs and the "fine grain" nature of the CFD applications.

In the case of one very-regular spectral global weather model, the sustained processor utilization dropped from the range of 40 percent – 45 percent on a 6 processor C90 to between 10 percent – 15 percent on a 60 processor SGI Origin 3000. In a more complicated meso-scale model that uses nested grids, comparative utilization figures from the C90 to the Origin 3000 were not available. However, this code maintained both the legacy OpenMP and more current MPI versions in a single code base, so such comparisons could be supported on current alternatives. In a wave modeling case the MPI port failed because, according to the researcher, multiple resolution grid nesting resulted in load balancing problems. This wave forecasting code runs operationally using OpenMP on a 4-to-8 processor Cray SV1. Research in this case is conducted on an Origin 3000 using shared memory and OpenMP, where performance flattens at around 24-32 processors.

However, the scaled-problem speedup of the majority of the new MPI codes is quite good. Rules of thumb like "use 200 grid points per processor" indicate how to scale the problem size as the number of processors increase to maintain performance. There is no doubt that available commodity HPCs will deliver the processing rates on the new MPI codes to satisfy the current operational forecast latency requirements. Also, the switchover to scalable and ever-improving commodity HPCs will enable the inclusion in the operational system of computationally-expensive model enhancements that could not be incorporated operationally because of the recent unavailability of state-of-the art vector supercomputers. Hence, the overall assessment for this area is "green."

As researchers have worked with the new distributed-memory programming approach, their "comfort level" has increased. However model enhancements will mean evolving these MPI codes, and this will continue to require investment by the weather and ocean modeling researchers in computer science activities rather than their science (the yellow "dilution of research" issue). Some of the user and developer groups interviewed get help by collaborating with experts at HPC centers, but they do not have the manpower or skill

base to do anything extra (e.g., software configuration management or maintaining hybrid OpenMP and MPI codes) that would enhance future software flexibility.

Another data point that bears on the suitability of current commodity HPCs is the oft-cited assessment that the “Europeans are five years ahead” in the weather forecasting area. In more concrete terms, the accuracy of the European 3-day typhoon forecast is gauged to be equivalent to our 2-day forecast. While not the only factor, European access to modern (Japanese) vector supercomputers has given them considerably more computing power over the last four years, and the relative ease of programming has meant the European weather researchers have been concentrating more on their weather science and less on computer science. This lack of state-of-the-art results could have potential negative national security impact, and as an isolated issue is judged to be in the “red” category. This is one example of the research “opportunity cost” of the current state of computing affairs for US national security researchers.

Planning activities for dispersion of airborne/waterborne contaminants also involves CFD. The models one group is employing use unstructured adaptive grids, and this significantly increases the complexity of the programming. The overall assessment is “green” because commodity HPCs are providing useful results to planning activities for military operations, intelligence gathering, counter-terrorism, and treaty monitoring. In one planning case there is a 10-hour timing requirement, which commodity HPCs are meeting.

As is common with such CFD applications, a larger commodity HPC would not necessarily produce useful speedup on a fixed problem size. Thus, if the scenario switched from planning to responding to an actual terrorist event, the current models running on commodity HPCs would not help in a low-latency crises situation (hence the “red” assessment). What is clearly needed is a more efficient computing solution, and the research group involved is exploring access to Japanese vector supercomputers through Cray Inc.

Engineering design of aircraft, ships, and other structures involves both computational structural mechanics (CSM) and CFD. Groups interviewed are using CSM to design safer military and embassy buildings as well as to perform forensic analysis after terrorist bomb attacks. CFD is being used to model the flow fields around aircraft to augment wind tunnel experiments. All codes considered had been ported using MPI over a 2 – 3 year period, often by development groups other than the ultimate users.

Users of third party CSM/CFD MPI codes are major consumers of cycles at DOD HPCMP facilities. Their reliance on scaled-problem speedup for their research activities would fill up any computational resource for as long as they could gain access. As far as the users interviewed are concerned, commodity HPCs are providing the needed capability for their research and hence the “green” assessment. In fact, their most significant concern appeared to be the queuing delays currently encountered to gain access to available commodity HPCs.

However, the users of third party codes interviewed tended to have no idea how efficiently (in terms of sustained processor utilization) commodity HPCs run their CSM and CFD jobs, or what the potential for improving their research productivity would be if the codes could run more efficiently. One of the more sophisticated development groups interviewed had experience running their CFD codes on both current Japanese vector supercomputers and commodity HPCs and reported up to a 25 times difference in sustained processor utilization: 50 percent on a vector supercomputer versus 2 percent – 10 percent on a commodity HPC. Their codes use unstructured meshes and adaptive mesh refinements during run time. In addition, the use of complicated physics in multiple domains poses load balancing challenges for distributed memory. These developers, unhappy with the programming complexity of MPI and the performance of commodity HPCs, are actively seeking access to Japanese vector supercomputers.

The areas of **weapon (warhead/penetrators) effect studies** and **improved armor design** uses computational mechanics to understand complex projectile-target interactions. CFD is also used for modeling flight dynamics of missiles and projectiles. Again, the vast majority of these applications have been ported using MPI (by third parties) to run on commodity HPCs. Again, these users are big consumers of cycles at DOD HPCMP facilities. One research group reported that their “grand challenge” computations take months to run on commodity HPCs. So although the overall assessment is “green,” the same (“yellow”) issues of processing efficiency and research productivity occur, but magnified by the length of time these high-priority applications are allowed to run.

One group reported a CFD application that used shared memory and OpenMP on symmetric multiprocessors and the Origin 3000 with up to 64 processors. The use of scalable shared-memory commodity HPC technology to defer the switch over to more complicated message passing is something to consider today. This approach was successful in this case in part because the group had an expert programmer who spent considerable effort to optimize the sequential version of the code on cache-based microprocessors. Not every research group has such a “guru.”

Because their global-memory bandwidth is limited, commodity HPCs today do not provide the needed capability for **cryptanalysis** (hence the “red” assessment). This same conclusion was reached previously by a recent DSB Task Force. Quoting from the *Report of the DSB Task Force on DOD Supercomputing Needs, 11 October 2000*:

The Task Force determined that the cryptanalysis application domain has a critical requirement for HPCs with high-random-access-global-memory bandwidth. There are three dimensions to this computing requirement:

- (1) the rate of random access to global memory measured in billions of updates/second (GUPS)
- (2) the size of the global memory, and
- (3) the ease of programming.

The first two dimensions translate directly into application capability. The third dimension bears on how easy it is to actually apply the computing capability. In the case of research activities involving a domain expert, even one with significant computer science skills, a difficult programming environment can eliminate an otherwise capable system from consideration. Ease of programming is also important for operational uses, but it usually does not represent a “show stopper” since application programs can be built to specification by a team of expert programmers.

The ease of programming factor is of concern due to the “dilution of research” issue (“yellow”) mentioned previously.

In contrast to applications involving CFD and CSM, the area of **survivability/stealthiness design** is distinguished by how efficiently commodity HPCs are working. This area of computational electro-magnetics (CEM) uses a variety of solution approaches. One popular approach uses the “method of moments” and reduces the computation to solving a dense system of linear equations. This is the same computational kernel as the LINPACK benchmark used in assessing the “Top 500” HPCs. There are both shared-memory and distributed-memory implementations that deliver upwards of 70 percent – 80 percent sustained processor utilization on both vector supercomputers and commodity HPCs for large-sized problems. This order n^3 algorithm has a high computation to communication ratio, and techniques exist to hide the latency from the processor to memory. The only issue (assessed as “yellow”) is that current DARPA research has produced a more efficient computational method for this application and the implementation impact on processor efficiency is yet to be fully assessed.

Commodity HPCs are used effectively today in many operational **intelligence, surveillance, and reconnaissance (ISR)** applications as well as by the signal and image processing research community to develop new exploitation techniques like automatic target recognition. Many of these operational applications are embedded, for example, forming synthetic aperture radar (SAR) images either at a ground station in the field or on an airplane such as the U2 reconnaissance platform. As a result, this HPC application community has naturally focused on the efficiency of its processing.

The history of U2 SAR image formation processing by one defense integrator is representative. When the SAR image formation code was ported from a Cray YMP to an SGI Power Challenge (MIPS R8000 processor), the eight-processor utilization dropped from 50 percent to 16 percent. On the Origin 2000 containing the 195 MHz MIPS R10000 processor with a 2/3 speed cache, the sustained processing utilization improved to 25 percent. The 250 MHz MIPS R10000 with a full speed cache resulted in a further utilization improvement to 29 percent. These applications use well-structured dense linear algebra algorithms with vector sizes that allow multiple vectors to fit into cache. Vendor optimized libraries for functions such as the fast Fourier transform are used to increase processing efficiency.

The issue identified as “yellow” in Figure 2 concerns the scale of some current and future ISR applications. In some of today’s largest applications, special-purpose accelerators must be combined with commodity HPCs to meet the operational processing throughput requirements. This is expensive, but reduces the overall computer installation to a practical size. Switching to all commercial processing is a future goal.

The group interviewed associated with the **national missile defense** area provides scientific and technical intelligence to BMDO. Applications included assessing radar and infrared signatures, computational aerodynamics, and modeling and simulation. Commodity HPCs are providing the needed capability with many applications requiring only a single processor or networks of loosely coupled workstations or small servers. The author is also aware that the radar signal processing and control involved in missile defense applications is being hosted on shared-memory commodity HPCs.

Finally, the group interviewed associated with the **test and evaluation** area effectively uses commodity HPCs to implement models and simulations and to drive visualizations for human-in-the-loop tests. Shared-memory is used with explicit threads (i.e., not OpenMP). Tests require real-time updates and special reflexive memory technology is used to support predictable shared-memory over a network. The only issue (“yellow”) is that currently special-purpose sensor emulators are used and there could be potential cost reductions and increased flexibility if these were replaced by HPCs.

6. GENERAL FINDINGS

This section summarizes the general findings of the survey.

1. Legacy applications have migrated from vector supercomputers to commodity HPCs using OpenMP (easy) or Message Passing Interface (difficult).

There is currently widespread use of portable software technology for achieving parallelism by the national security community: Message Passing Interface (MPI) for distributed memory and OpenMP for shared-memory multitasking. This is insulating application codes from the vagaries of the HPC marketplace.

The appearance of increasingly powerful commodity HPCs that support multi-tasking through OpenMP directives has made this approach attractive for a low to moderate number of processors. Many users can quickly get speedup through the addition of OpenMP compiler directives. This is providing an easy migration path from legacy vector supercomputers. Support staff at the HPCMP facilities are using this approach to cultivate new users. Finally, more sophisticated users are also programming shared-memory systems with portable thread packages.

In contrast, moving from the shared-memory to the distributed-memory paradigm has required multiyear software efforts involving significant investments. The DOD HPCMP has supported a number of such conversions on the Common High Performance Computing Software Support Initiative (CHSSI) program. Other government organizations have done likewise (e.g., DOE and NASA).

2. Vector supercomputers process more efficiently than commodity HPCs.

There was a significant drop in sustained processor utilization when legacy vector supercomputer codes were run on commodity HPCs because of the memory-processor bottleneck of current microprocessors. Order of magnitude drops in the percentage of peak processing rates were not uncommon: “50 percent to 5 percent.” Code restructuring (to make compatible with caching) recovered only some of this loss for most applications.

The efficiency of the “big-hitter” production codes being run at DOD HPC facilities is difficult to assess because sustained processor utilization is not widely known. Most of the users of third-party codes and many of the developers surveyed had no idea of their code’s efficiency. The user community appears to be generally unaware of just what the potential is for improving the value of the computing resources they use. Anecdotes indicate that sustained processor utilization might be as low as 2 percent for some complicated CFD or CSM applications.

For the sake of illustration, a modern 16-processor shared-memory parallel vector supercomputer (peak 10 Gflop/s vector processors with 256 GB of memory) programmed with compiler directives delivers 80 Gflop/s at a sustained processor utilization of 50

percent. This is equivalent to the sustained performance of an *800-processor* distributed-memory commodity HPC (peak 1 Gflop/s microprocessors with 400 GB of memory) programmed with message passing that is sustaining 10 percent of peak. The total cost of ownership of these systems includes factors such as the programming complexity, power, reliability, and space requirements in each case. Tradeoffs between these two alternatives have not been made recently in the US because state-of-the-art vector supercomputers have been unavailable over the last four years.

3. Most (big) applications scale well on commodity HPCs—provided the problems decompose well.

To maintain performance for many applications, the size of the application must be increased as the machine size is increased. This approach of scaled-problem speedup is compatible with research applications where more resolution (hence bigger problem size) is better. It is less applicable to operational situations where the point is to drive down the time to solution for a fixed problem.

Parallel efficiencies based on scaled-problems speedup as high as 80 percent are often attained on both vector supercomputers and commodity HPCs for applications that involve straightforward domain decomposition methods. This measure of “efficiency” should not be confused with the level of sustained performance discussed above. If single sustained processor utilization is 10 percent, then a system with parallel efficiency of 80 percent is delivering 8 percent of the total peak processing potential of the machine.

Because of all the current MPI codes and their applicability to large problems, commodity HPCs at DOD HPC facilities are a scarce resource, and a significant issue is the queuing delays required for access. Researchers routinely reduce the number of processing nodes requested to increase overall response time.

4. Some applications scale poorly on commodity HPCs.

Applications that must access global memory in an irregular and unpredictable fashion have the most trouble on today’s commodity HPCs. This kind of processing is essential for cryptanalysis research. Computational fluid dynamics that use domain decomposition where the grids or meshes are unstructured or are adapted during run time pose both programming complexity and scaling issues. Some grid/mesh adaptations currently occur on a single processor in the middle of a parallel run. Computational structural mechanics involving complex dynamics such as when there are a lot of small pieces flying around pose performance challenges. Finally, models that contain a lot of non-uniform physics result in complicated load balance requirements.

5. Negative impact on research output in some cases—biggest loser is domain researcher who is forced to program distributed-memory HPCs

The recent lack of a state-of-the-art vector supercomputer option in the United States has meant entire scientific communities have had to switch their processing to distributed-

memory commodity HPCs. The difficulty of message passing programming has meant that researchers spend considerably more time than in the past on computer science rather than on their science. The dilution of research and the opportunities that are being lost are difficult to quantify across the board—the weather forecasting community seems to have the clearest case in this regard. But, most scientific communities can correlate past breakthroughs with increases in computing power, and so a situation where computing tools are difficult to use or where they can only be used inefficiently is not a desirable state of affairs.

7. RECOMMENDATIONS

This section concludes with recommendations.

1. Assess the usefulness of Japanese vector supercomputers (first for R&D applications).

The national security community should assess the usefulness of Japanese vector supercomputers. Their potential for delivering higher processing efficiency with less programming complexity addresses a number of issues identified by this survey, not to mention the potential upside of any scientific breakthroughs that result. Price/performance tradeoffs that include the total cost of ownership should be the guide in making acquisition selections. The competition that this promotes could have the positive effect of focusing US computer vendors on improving the efficiency of their systems. The place to start is with non-mission-critical research applications that are characterized by long lead times where the impact of a future lack of foreign availability is less severe.

2. Reach out to researchers through the use of OpenMP on shared-memory systems—use MPI on distributed-memory systems as next resort.

The capability of today's shared memory systems is such that they provide a substantial entry-level HPC resource. For codes that contain loop level parallelism, speedup can be easily achieved through the inclusion of OpenMP compiler directives. This is the recommended first step before resorting to a more involved effort of switching to distributed-memory message passing. The availability of more capable shared-memory vector supercomputers could reduce the number of MPI conversions needed.

3. Promote flexibility through software that combines OpenMP and MPI and that switches between vector and cache-based optimizations.

The most sophisticated development groups encountered during the survey have maintained a single code base as the computing options have evolved. Today their codes can be compiled to use OpenMP directives and/or MPI. Indeed, emerging clusters of shared-memory servers suggest hybrid strategies that use both approaches during a single run. In addition, these groups have maintained optimizations for both vector and cache-based microprocessors in their code base. Such flexible coding strategies should be encouraged as they provide the most options to respond to future developments in the HPC marketplace. A good way to promote this is to provide funding to augment the less computationally sophisticated research groups with computational science experts.

4. Establish a multifaceted R&D program to improve the *productivity* of high performance computing for national security applications.

The national security community needs computers that work better than the commodity HPCs it currently is using. Improvements in sustained processor utilization would make current-sized HPCs more useful. Simpler software development approaches would make

researchers and developers of operational systems more productive. Improvements in both system performance and software productivity are needed without sacrificing the current gains that have been made in software portability. A multi-faceted R&D program focusing on algorithm, architecture, and software improvements is needed.

As a first step it is essential that a comprehensive assessment of current national-security HPC software be conducted. This is needed to establish goals for improvement in processing efficiency and to make wise investment decisions. Also, actual national security research and operational applications of HPC should be selected to demonstrate the impact of such an R&D program. These should be a combination of existing and future applications with stressful HPC requirements.

In conclusion, the bottom line for the national security community reduces to the interrelated issues of *productivity* and *affordability*. How productive are the researchers/developers that write the high performance software? How productive (efficient) are the HPCs that run the software? What does it cost? Included in the total cost are the facilities and the operational risks associated with the reliability of larger less-efficient installations. Perhaps the hardest cost to quantify are the “opportunities lost” when domain researchers spend their time on complicated computer programming rather than on their science.

APPENDIX A

INTERVIEW QUESTIONS

Date:

Name:

Title:

Employer:

Phone:

Email:

Sponsor Name:

Sponsor Affiliation:

1. Personal Background

1.1. What is your current role?

1.2. Length of time in current position?

1.3. Length of time involved with high performance computing?

2. HPC Application Description

2.1. Describe your application. (Characterize it as: research and development, test and evaluation, operational system, other.)

2.2. What is the national security significance of your application?

2.3. What are current “mission” requirements (e.g., problem size, timing) that force the use of HPCs?

2.4. What are the future “mission” requirements (e.g., problem size, timing) that force the use of HPCs?

3. Software Description

3.1. What is the name of the software you are using/developing?

3.2. Who or what organization developed the code you are using?

3.3. What are the inputs?

- 3.4. What are the outputs?
- 3.5. What algorithm(s) is (are) implemented by the software?
- 3.6. Are there vectors involved?
- 3.7. How do you measure the computational complexity of the algorithm(s)?
- 3.8. If the software runs on more than one processor, how is parallelism achieved at the algorithmic level?
- 3.9. Is the algorithm data-parallel?
- 3.10. Are your computational problems amenable to domain decomposition (can the problem and memory be partitioned?) or are they dependent upon data located across the entire computational domain or problem space?
- 3.11. What programming language is the software currently written in? In the future?
- 3.12. Are you satisfied with the performance of the compiler? How do you measure the compiler's performance?
- 3.13. Do you use a shared or distributed-memory programming model? Or a combination of the two?
- 3.14. Is a compiler or are compiler directives used to parallelize the code? If so, do you use or are you considering the use of OpenMP?
- 3.15. If message passing, do you use MPI? Two sided (e.g., send/receive)? One sided (e.g., put/get)?
- 3.16. If message passing, do you use MPI collective operations or other previously developed support for global data reorganizations (e.g. distributed matrix transpose)?
- 3.17. How would you characterize the communication "granularity" of your parallel application (fine or coarse "grain")? Explain.
- 3.18. How would you characterize the synchronization "granularity" of your parallel application (fine or coarse "grain")? Explain.
- 3.19. Do you use computational libraries? (E.g., vendor-proprietary math/signal processing, LAPACK, SCALAPACK, ...).
- 3.20. Besides compilers and computation/communication libraries, what tools do you use to develop and debug your code? Are you satisfied with these tools?

- 3.21. How much time (roughly in percentages) do you spend on: (1) initial design, (2) initial coding, (3) debugging, and (4) performance optimization.
- 3.22. How long does it take to develop your application software—the time to initial useful results (roughly: days, weeks, months, years)?
- 3.23. Discuss the importance of ease-of-programming to your effort and the impact of this factor on the selection of HPC architecture.

4. Typical Performance

- 4.1. On what computers do you run the software?
 - What is the processor clock speed?
 - What is flop/s rating of each processing node?
- 4.2. How many processors (or processing nodes) are typically used?
- 4.3. What is the largest number of processors ever used successfully on a single problem instance?
- 4.4. What is the size of the input data?
- 4.5. What is the size of the output data?
- 4.6. Is problem size a variable (“bigger always better”) or does your mission application fix it?
- 4.7. How long does a typical run of the software take for a problem instance?
- 4.8. Discuss the importance of time-to-solution in the use of your software.
- 4.9. Does some version of the problem run on a single processor? In core? Out of core?
- 4.10. What is the utilization of the floating point unit(s) on a single processor?
- 4.11. Have you benchmarked the software on an increasing number of processors?
- 4.12. What kinds of parallel speedups do you achieve?
- 4.13. Is there a point where performance plateaus? Gets worse?
- 4.14. What are the algorithmic issues that inhibit scalability and performance (e.g., surface to volume ratio of domain, irregular accesses to global memory, etc.)?

- 4.15. Do you have benchmark results that compare versions of the code that use different programming approaches on the same machine? (E.g., running a message passing version on an SMP for which you also have compiler-based results.)

5. Portability and Porting

- 5.1. What architectural classes of HPCs does your software run on? (SMPs, DSM, MPPs, vector, parallel vector, scalable vector, commodity clusters, high-end clusters, etc.)
- 5.2. Are you contemplating a port to a new architectural class in the future? For example from shared memory to a distributed memory/cluster architecture?
- 5.3. In these porting exercises, is the same application software just recompiled (e.g., using message passing on an MPP and an SMP)? Or are their different versions of the software (including even different algorithms) for different HPC types?
- 5.4. If the algorithm has changed because of porting activities, please explain.
- 5.5. If you have been involved in porting activities, how long have they taken (days, weeks, months, years). Distinguish between porting *between* or *within* architectural classes.
- 5.6. Do you have benchmark results that compare the performance of the software between architectural classes? If so, what are your conclusions?

6. Current Architectural Limitations

- 6.1. What are the architectural features of an HPC system that have the greatest impact on the performance of your software? Why?
- 6.2. Do your algorithms have data-level parallelism but limited data reuse (yielding poor cache performance in microprocessors)?
- 6.3. Do your algorithms require irregular access to a large amount of data (need zero-wait state, non-cached memory)?
- 6.4. What architectural improvements would have the biggest impact on performance. For example:
- Processor floating-point speed
 - Processor integer speed
 - Cache size
 - Processor access to “local” memory (latency and bandwidth)
 - Processor access to “global” memory (latency and bandwidth)
 - Uniform memory access time across global memory

- Amount of memory per processing node
- Interconnect latency
- Interconnect bandwidth
- Interconnect topology
- Interconnection network interface directly to system bus (vs. I/O bus)
- Access (latency and/or bandwidth) to secondary storage

7. Documentation

What documentation is available on your software, development techniques, processing performance, and verification process used to confirm the validity of the approach?

8. Additional Applications

8.1. In addition to the work cited above, what other HPC applications are envisioned using the processing techniques being developed by you or your group? Are there any applications outside the national security area? Government? Commercial?

9. Additional Sources

- 9.1. Who do you know that really understands the interplay between HPC architecture and the performance of their application code?
- 9.2. Who do you know that has ported their application(s) across multiple HPC architectures
- 9.3. Who do you know that has benchmarked their application code across different HPC architectures?

APPENDIX B

BACKGROUND INFORMATION OF APPLICATION ASSESSMENTS

This appendix contains additional background information on each of the 10 application areas surveyed:





1. Operational weather and ocean forecasting
2. Planning activities for dispersion of airborne/waterborne contaminants
3. Engineering design of large aircraft, ship, and structures
4. Weapon (warheads and penetrators) effect studies
5. Improved armor design
6. Cryptanalysis
7. Survivability/stealthiness design
8. Intelligence, reconnaissance, and surveillance
9. National missile defense
10. Test and evaluation

For each application area, a quad chart is presented describing the organizations and number of people interviewed, the national security significance of the application, details on the software discussed, and finally the assessment with individual issues itemized. In the assessments, green indicates that commodity HPCs are providing needed capability. Yellow indicates that there are significant issues with the use of commodity HPCs, but generally they are providing the needed capability. Red indicates a serious problem with the use of commodity HPCs.




Weather and Ocean Forecasting

<p>Organizations (# people)</p> <ul style="list-style-type: none"> ● Fleet Numeric Meteorology and Oceanography (FNMOC), Monterey, CA (3) ● Naval Meteorology & Oceanography Command, Stennis, MS (1) ● Naval Research Laboratory, Monterey, CA (8) ● Naval Research Laboratory, Stennis, MS (1) ● Army Corp of Engineer R&D Center, Vicksburg, MS (1) ● Naval Research Laboratory, Wash DC (2) 	<p>National Security Significance</p> <ul style="list-style-type: none"> ● FNMOC and NAVOceanO provide operational weather and ocean wave forecasting to services multiple times a day on a daily basis ● For example, accurately predicting the course of typhoons has safety and cost implications for the surface Navy ● NRL and ACERDC do the research and development of the models that feed the operational systems
<p>Software</p> <ul style="list-style-type: none"> ● NOGAPS global weather prediction uses MPI-2 put/get. Processor utilization of ~40% on C90s has dropped to ~14% on SGI O3K, (6 proc. C90 = 20 proc. O3K), scales well ● COAMPS regional weather prediction is a more complicated nested grid model. Use a hybrid OpenMP and MPI approach, although MPI on O3K is most used/tested ● WAM wave modeling uses OpenMP on SV1 and O3K (MPI failed). Scales to 24 nodes. 	<p>Assessment</p> <ul style="list-style-type: none"> ● G FNMOC is replacing 16 proc. and 8 proc. Cray C90s with 128 proc. SGI O3K in May 2001, add'l 512 proc. O3K summer 2001 ● Y Dilution of effort: weather researchers now content, but transitioning from vector to commodity HPCs was a difficult multi-year undertaking--evolving will continue to cost ● R Results are not state of the art--commodity HPCs put USA at a disadvantage relative to Europeans--"five years behind"

Dispersion of Airborne/Waterborne Contaminants

<p>Organizations (# people)</p> <ul style="list-style-type: none"> ● SAIC Center of Atmospheric Physics, McLean, VA (4) ● Army Corp of Engineer R&D Center, Vicksburg, MS (1) 	<p>National Security Significance</p> <ul style="list-style-type: none"> ● Predict the dispersion of hazardous aerosols and gasses in the atmosphere ● Support military operation planning and execution, intelligence gathering, counter terrorism, and treaty monitoring ● Support DTRA and SOCOM ● Capability originally developed in response to Gulf War studies
<p>Software</p> <ul style="list-style-type: none"> ● OMEGA regional weather prediction software uses adaptive unstructured horizontal grid, sequential adaptation is performed 10-15 times per 10 hour run. Structured vertical grid vectorizes ● Uses MPI with domain decomposition ● Irregular accesses to memory means sensitive to cache versus memory size in local domain--communication overhead limits scaling 	<p>Assessment</p> <ul style="list-style-type: none">  Ease of programming important to this group, but code complexity meant OpenMP performed badly and so switched to MPI  Run on 16 proc. O2K and SV1 to meet 10 hour planning cycle; to scale problem would only require bigger commodity HPC  Would not be able to support a low-latency crises requirement at current resolution  Following the SV-2 for DTRA. Talking to Cray about access to NEC SX-5

Engineering Design of Structures

<p>Organizations (# people)</p> <ul style="list-style-type: none"> ● Army Research Lab, Aberdeen, MD (1) ● NAVAIR, Pax River (2) ● SAIC Laboratory for Applied Computational Science, McLean, VA (2) ● George Mason University, Fairfax, VA (1) 	<p>National Security Significance</p> <ul style="list-style-type: none"> ● Computational structural mechanics used to do forensic analysis after terrorist bomb attacks and predictive analysis for the design of safer military and embassy structures ● Computational fluid dynamics model flow fields around complete aircraft--augment wind tunnel experiments to reduce costs ● Providing acquisition support for recce pods on the F/18 fighter program.
<p>Software</p> <ul style="list-style-type: none"> ● PARADYN CSM unstructured grid, MPI: 90% of applications give almost linear scaled speedup up to 512 processors. However many small fragments flying around require global communications and causes less than linear speed up for the other 10% ● FEFLOW CFD adaptive mesh, MPI or OpenMP, optimized for vector or cache, processor utilization: 2%- 10% on commodity HPCs versus 50% on vector 	<p>Assessment</p> <ul style="list-style-type: none">  Users of third party CSM/CFD MPI codes are major consumers of cycles at HPCMOD facilities, limited by access/queuing delays  Developers of complex code unhappy with programming complexity of MPI and performance of commodity HPCs; Seeking <i>foreign</i> access to Japanese vectors  All codes discussed have been converted through domain decomposition to use MPI over 2 - 3 year period

Weapon (Warhead/Penetrators) Effects Studies

Organizations (# people)

- Army Research Lab, Aberdeen, MD (3)



National Security Significance

- Computational mechanics used to understand complex projectile-target interactions to develop advanced survivability and *lethality* technologies.
- Computational fluid dynamics for modeling flight dynamics of missiles and projectiles
- Use computer models to augment experimentation to reduce costs and explore new concepts that would be difficult to test

Software

- CTH multi-dimensional multi-material finite volume shock wave propagation CM code from Sandia; structured 3D mesh, uses domain decomposition and MPI. Always scale problem size to fill biggest machine they can get time on. Can run for months!
- Zonal Navier-Stokes Flow (ZNSFlow) CFD. Structured multi-block. Uses shared memory and OpenMP on SMPs and SGI O2K/O3K up to 64 processors. Much work optimizing sequential version.

Assessment

-  Users of third party CM MPI codes are major consumers of cycles at HPCMOD facilities, limited by access/queuing delays
-  In general, users of third party codes running on commodity HPCs lack knowledge of their efficiency and how much “grand challenge” computations that are taking *months* could be shortened by more compatible vector architectures that would deliver higher processor utilization

Improved Armor Design

Organizations (# people)

- Army Research Lab, Aberdeen, MD (2)



National Security Significance

- Computational mechanics used to understand complex projectile-target interactions to develop advanced survivability and lethality technologies.
- Use computer models to augment experimentation to reduce costs and explore new concepts that would be difficult to test

Software

- CTH multi-dimensional multi-material finite volume shock wave propagation CM code from Sandia; structured 3D mesh, uses domain decomposition and MPI. Always scale problem size to fill biggest machine they can get time on. Can run for months!

Assessment

-  Users of third party CM MPI codes are major consumers of cycles at HPCMOD facilities, limited by access/queuing delays
-  In general, users of third party codes running on commodity HPCs lack knowledge of their efficiency and how much “grand challenge” computations that are taking *months* could be shortened by more compatible vector architectures that would deliver higher processor utilization

Cryptanalysis

Organizations (# people)

- National Security Agency, Fort Meade, MD (8)




National Security Significance

- Decrypting secrecy codes that are used to hide information over various transmission channels
- Both an R&D and operational aspect to this activity

Software

- Combination of small quickly evolving exploratory software programs for research and highly optimized utility programs for research and operations
- Often researchers will use a single processor and all the memory available. OpenMP on shared memory preferred
- MPI is not used for distributed memory because of performance. Use vendor shmem library on T3E. Currently exploring the use of IDA's UPC extensions of C

Assessment

-  Need systems that provide extremely fast access to extremely large global memories--synonymous in the past with vector supercomputers
-  Any programming model other than shared memory will likely result in dilution of research effort--difficult to quantify impact
-  There are a limited number of cryptanalytic apps that require no network traffic and so are compatible with commodity HPCs

Survivability/Stealthiness Design

Organizations (# people)

- Air Force Research Laboratory, Dayton, OH (1)
- Northrop Grumman, Los Angeles, CA (1)



National Security Significance

- Computational electromagnetics for radar cross-section/signature prediction
- Perform research into reducing the radar signatures of airplanes such as the JSF and F22
- Provide technical support to acquisition activity

Software

- XPATCH code does ray tracing and can be parallelized perfectly. Not an HPC capability problem
- SWITCH example of method of moments approach that reduces to directly solving dense systems of linear equations
- This can be done *extremely* well on commodity HPCs (cf. Top 500 benchmark) using highly optimized MPI or threads: 90% processor utilization in core and 80% *out of* core on the SGI O3K for example!

Assessment

-  The current approach to this application reduces to implementing a highly structured $O(n^3)$ algorithm which can be computed extremely well on all available HPC architectures
-  Current DARPA emphasis is on developing new fast $O(n^2)$ iterative solution algorithms that are less memory intensive. This will allow substantial increases in model resolutions but raise numerical and architectural issues

Intelligence, Surveillance, and Reconnaissance

Organizations (# people)

- Northrop Grumman, Baltimore, MD (1)
- Signal processing site (4)



National Security Significance

- Processing the outputs of various types of sensors to produce battlespace situation awareness or other actionable intelligence
- Target cueing, aided target recognition, and other special exploitation products
- These operational applications have to meet throughput and latency requirements as part of a larger system
- There also may be size, weight, power constraints

Software

- U2 synthetic aperture radar image formation ported from a Cray YMP to SGI Power Challenge (R8000), 8 processor utilization dropped from 50% to 16%. The 195MHz R10000 with a 2/3 speed cache improved this to 25%. The 250MHz R10000 has a full speed cache and the utilization is now 29% (All use vendor optimized FFT)
- This software became the Common Imagery Processor that is deployed at tactical ground stations. Uses OpenMP.

Assessment

-  Well structured linear algebra algorithms with vector sizes that allow multiple vectors to fit into cache result in above average processor utilization on commodity HPCs. Some limited global communication required--not as significant a factor
-  Processing requirements can get very large in certain instances and the fallback today is to add special purpose accelerators to commodity HPCs. Possible application for scalable vector machine if affordable

National Missile Defense

Organizations (# people)

- Missile, Space, Intelligence Command, Huntsville, AL (5)


National Security Significance

- Provide scientific and technical intelligence to a variety of customers including BMDO
- Understand the performance of and assess the threat of surface-to-air missiles, ballistic missiles, anti-tank guided systems, directed energy weapons
- Applications include RF signatures, IF signatures, computational aerodynamics, and modeling and simulation (one-on-one engagements)

Software

- By and large MSIC uses their HPC resources for throughput. Jobs are scheduled onto single processors. They mostly use their 4-processor SV1 in this fashion. They don't support MPI.
- TIGER CFD code to be applied to study spinning missile tails in the future. OpenMP and can switch between cache and vector optimizations. Steady state solution. Scales up to 12 proc. Utilization for 8 proc.: SV1: O2K = 5.6 : 1; SV1:SUN420RS = 3.7 : 1

Assessment

-  MSIC current and future HPC requirement appear to be easily met with commodity HPCs, in fact with an inexpensive LINUX cluster (e.g., to run XPATCH)

Test and Evaluation

Organizations (# people)

- Naval Air Warfare Center, Pax River (2)



National Security Significance

- Modeling and simulation as part of test and evaluation
- Computer simulation as part of a integrated test facility containing sensor emulators, human interfaces, and a system under test
- Reducing the cost of test and evaluation through HPC plus simulating test scenarios that would be difficult to replicate in reality

Software

- Use shared memory commodity HPCs to drive visualization for human-in-the-loop simulations
- Use threads rather than OpenMP
- “Real time” application and special reflective memory support (SCRAMNET) is used
- NUMA on Origin 2000 can introduce jitter if threads are not pinned to processors.

Assessment

-  Commodity HPCs providing needed modeling, simulation, and visualization support
-  Could potentially reduce costs further and increase flexibility by replacing sensor emulators with programmable HPCs

APPENDIX C

STUDY MEMORANDUM



OFFICE OF THE SECRETARY OF DEFENSE

1000 DEFENSE PENTAGON
WASHINGTON, DC 20301-1000



MAR 23 2001

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
DIRECTOR, BALLISTIC MISSILE DEFENSE ORGANIZATION
DIRECTOR, DEFENSE INTELLIGENCE AGENCY
DIRECTOR, DEFENSE THREAT REDUCTION AGENCY
DIRECTOR, NATIONAL IMAGERY AND MAPPING AGENCY
DIRECTOR, NATIONAL RECONNAISSANCE AND MAPPING
AGENCY
DIRECTOR, NATIONAL SECURITY AGENCY

SUBJECT: Survey of High Performance Computing Architecture Requirements

The Department of Defense is analyzing requirements of various features of high performance computing architectures that address our national security software applications. We would like to conduct interviews with key software developers and users of high performance computing within your organizations. Each Service and Agency is asked to provide a point of contact to facilitate these interviews. These interviews will be conducted at the appropriate security level to analyze user requirements.

Dr. Charles J. Holland will be our point of contact for this activity. Please email or phone your responses to him by March 28, 2001 at charles.holland@osd.mil, 703-695-0598. We expect these interviews to occur during the period of April 2 through April 13, 2001.

Dave Oliver
Principal Deputy Under Secretary
of Defense (AT&L)

Linton Wells II
Principal Deputy Assistant Secretary
of Defense (C3I)